

IN THE CLAIMS

1 1. (Previously Presented) A method for managing a code sequence, comprising:
2 determining first intermediate correlation values for a first plurality of sample sequences
3 to load in a plurality of sample sequence registers during a first clock cycle;
4 determining second intermediate correlation values for the first plurality of sample
5 sequences during a second clock cycle;
6 determining correlation outputs for the first plurality of sample sequences from the first
7 and second intermediate correlation values; and
8 determining a synchronization point that identifies an amount of delay incurred from
9 transmission of the sample sequences from the correlation outputs.

1 2. (Original) The method of Claim 1, wherein determining the first intermediate
2 correlation values comprises processing coefficients in a first code sequence group in parallel
3 with corresponding sample values in corresponding sample sequence groups from the first
4 plurality of sample sequences.

1 3. (Original) The method of Claim 1, wherein determining the second intermediate
2 correlation values comprises processing coefficients in a second code sequence group in parallel
3 with corresponding sample values in corresponding sample sequence groups from the first
4 plurality of sample sequences.

1 4. (Original) The method of Claim 1, wherein determining correlation outputs for the
2 first plurality of sample sequences comprises taking a sum of the first and second intermediate
3 correlation values for each of the first plurality of sample sequences.

1 5. (Original) The method of Claim 1, further comprising:
2 determining first intermediate correlation values for a second plurality of sample values
3 during a third clock;
4 determining second intermediate correlation values for the second plurality of sample
5 values during a fourth clock; and
6 determining correlation output values for the second plurality of sample value from the
7 first and second intermediate correlation values.

1 6. (Previously Presented) A method for managing a code sequence, comprising:
2 processing a first group of coefficients in the code sequence, loaded in a plurality of code
3 sequence registers during a first clock cycle, with a first group of contiguous sample values,
4 loaded in a plurality of sample sequence registers, in a received sample to determine a first
5 intermediate correlation value during the first clock cycle;
6 processing a second group of coefficients in the code sequence, loaded in the plurality of
7 code sequence registers previously used for the first group of coefficients during a second clock
8 cycle, with a second group of contiguous sample values, loaded in the plurality of sample
9 sequence registers, in the received sample to determine a second intermediate correlation value
10 during the second clock cycle;
11 determining a correlation output from the first and second intermediate correlation
12 values; and
13 determining a synchronization point that identifies an amount of delay incurred from
14 transmission of the sample values from the correlation output.

1 7. (Previously Presented) The method of claim 6, wherein the code sequence comprises
2 L coefficients and the first and second group of coefficients in the code sequence each comprises
3 n coefficients, where L and n may be any value.

1 8. (Original) The method of claim 7, wherein the first and second group of sample
2 values in the received sample each comprises n sample values.

1 9. (Original) The method of claim 6, wherein the first and second group of coefficients
2 in the code sequence are contiguous.

1 10. (Canceled)

1 11. (Original) The method of claim 6, wherein processing the first group of coefficient
2 in the code sequence with the first group of sample values in the received sample comprises
3 determining a sum of the products of the first group of coefficients in the code sequence with the
4 first group of sample values in the received sample.

1 12. (Original) The method of claim 6, wherein processing the second group of
2 coefficients in the code sequence with the second group of sample values in the received sample

3 comprises determining a sum of the products of the second group of coefficients in the code
4 sequence with the second group of sample values in the received sample.

5 13. (Original) The method of claim 6, wherein determining the correlation output from
the first and second intermediate correlation values comprises taking the sum of the first and
second intermediate correlation values.

1 14. (Previously Presented) A method for managing a code sequence, comprising:
2 organizing the code sequence, having L contiguous coefficients, into L/n contiguous
3 code sequence groups having n coefficients each, wherein n and L are greater than 1;
4 selecting a number of sample sequences to process in parallel where each of the sample
5 sequences has contiguous sample values from a received sample;
6 organizing contiguous sample values from each of a first set of contiguous sample
7 sequences to process in parallel into a first set of contiguous sample sequence groups;
8 processing coefficients in each of the code sequence groups loaded into a plurality of
9 code sequence registers in parallel with corresponding sample values in corresponding sample
10 sequence groups from the first set of sample sequences loaded into a plurality of sample
11 sequence registers, where each of code sequence groups is processed during a different clock
12 cycle;
13 determining a correlation output for each of the sample sequences; and
14 determining a synchronization point that identifies an amount of delay incurred from
15 transmission of the sample sequences from the correlation output.

1 15. (Original) The method of Claim 14, further comprising:
2 organizing contiguous sample values from each of a second set of sample sequences to
3 process in parallel into a second set of contiguous sample sequence groups; and
4 processing coefficients in each of the code sequence groups in parallel with
5 corresponding sample values in corresponding sample sequence groups from the second set of
6 sample sequences, where each of the code sequence groups is processed during a different clock
7 cycle.

1 16. (Canceled)

1 17. (Previously Presented) The method of Claim 16, wherein determining a
2 synchronization point comprises determining a correlation output having a highest numerical
3 value.

1 18. (Previously Presented) The method of Claim 14, wherein a first sample value in a
2 first sample sequence includes a first sample value in the received sample and each consecutive
3 sample sequence includes a next contiguous sample value in the received sample as a first sample
4 value of the consecutive sample sequence.

1 19. (Original) The method of Claim 14, wherein processing comprises determining a
2 sum of the products of the coefficients in each of the code sequence groups with each of the
3 sample values in corresponding sample sequence groups from the first set of sample sequences.

1 20. (Original) The method of Claim 14, wherein the code sequence comprises a plurality
2 of L contiguous values.

1 21. (Original) The method of Claim 20, wherein the code sequence is organized into a
2 plurality of n code sequence groups.

1 22. (Original) The method of Claim 21, wherein a number, d, sample sequences are
2 selected to process in parallel where each of the sample sequences has L contiguous sample
3 values from the sample.

1 23. (Original) The method of Claim 22, wherein the first set of sample sequences is
2 organized into a plurality of contiguous sample sequence groups having n values each.

1 24. (Canceled)

1 25. (Previously Presented) The method of Claim 14, wherein processing coefficients
2 comprises processing coefficients for L/n clocks.

1 26. (Previously Presented) A method for managing a code sequence, comprising:
2 organizing the code sequence, having L contiguous coefficients, into L/n contiguous
3 code sequence groups having n coefficients each, wherein n and L are greater than 1;

4 selecting a number of sample sequences, d , to process in parallel where each of the
5 sample sequences has L contiguous sample values from a received sample, where a first sample
6 value in a first sample sequence is a first sample value in the received sample and each
7 consecutive sample sequence includes a next contiguous sample value in the received sample as
8 a first sample value in the consecutive sample sequence;

9 organizing sample values from each of a first set of d sample sequences into a first set of
10 sample sequence groups having n values each and loading the set into a plurality of sample
11 sequence registers;

12 processing coefficients in each of the code sequence groups loaded into a plurality of
13 code sequence registers in parallel with corresponding sample values in corresponding sample
14 sequence groups from the first set of d sample sequences, where each of the code sequence
15 groups is processed during a different clock cycle, where d may be any value;

16 determining a correlation output for each of the sample sequences; and

17 determining a synchronization point that identifies an amount of delay incurred from
18 transmission of the sample sequences from the correlation output.

1 27. (Previously Presented) The method of Claim 26, further comprising:

2 organizing sample values from each of a second set of d sample sequences into a second
3 set of contiguous sample sequence groups having n values each; and processing values in each of
4 the code sequence groups in parallel with corresponding sample values in corresponding sample
5 sequence groups from the second set of d sample sequences, where each of the code sequence
6 groups is processed during a different clock cycle.

1 28. (Canceled)

1 29. (Original) The method of Claim 28, wherein determining a synchronization output
2 comprises determining a correlation output having a highest numerical value.

1 30. (Original) The method of Claim 26, wherein the code sequence is organized into L/n
2 groups.

1 31. (Original) The method of Claim 26, wherein processing comprises determining a
2 sum of the products of the coefficients in each of the code sequence groups with each of the
3 sample values in corresponding sample sequence groups from the first set of d sample sequences.

1 32. (Original) The method of Claim 26, wherein the processing is completed after L/n
2 clocks.

1 33. (Previously Presented) A correlator unit, comprising:
2 a plurality of code sequence registers that store coefficients from a code sequence group
3 having n coefficients, the plurality of code sequence registers storing coefficients from one code
4 sequence group of L/n code sequence groups at a time, where L is the number of coefficients in a
5 code sequence, wherein n and L are greater than 1;
6 a plurality of sample registers that store sample values from a plurality of sample
7 sequences that are processed in parallel;
8 a processing unit that processes coefficients in each of the plurality of code sequence
9 groups in the plurality of code sequence registers in parallel with corresponding sample values in
10 corresponding sample sequence groups from a first plurality of sample sequences in the plurality
11 of sample registers, where each of the code sequence groups is processed to generate
12 intermediate correlation values during a different clock cycle;
13 an accumulation unit that generates a correlation output for each of the sample sequences
14 from the intermediate correlation values generated during the different clock cycles; and
15 a correlation output processor that determines a synchronization point that identifies an
16 amount of delay incurred from transmission of the sample sequences from the correlation output.

1 34. (Previously Presented) The correlator unit of Claim 33, wherein the accumulation
2 unit further comprises a plurality of accumulation sub-units each accumulation sub-unit receiving
3 results from the processing unit for a designated sample sequence, each accumulation unit
4 generating a correlation value for the designated sample sequence after each of the code
5 sequence groups are processed.

1 35. (Previously Presented) The correlator unit of Claim 33, wherein the processing unit
2 processes the coefficients in each of the plurality of code sequence groups in the plurality of code
3 sequence registers in parallel with corresponding sample values in corresponding sample
4 sequence groups from a second plurality of sample sequences in the plurality of sample registers,
5 where each of the code sequence groups is processed during a different clock cycle.

1 36. (Canceled)

1 37. (Original) The correlator unit of Claim 36, wherein the correlation output processor
2 determines a synchronization point from a correlation output having a highest numerical value.

1 38. (Previously Presented) The correlator unit of Claim 33, wherein the processing unit
2 determines a sum of products of the coefficients in each of the code sequence groups with
3 corresponding sample values in corresponding sample sequence groups.

1 39. (Previously Presented) A correlator unit, comprising:
2 a plurality of n code sequence registers that store n coefficients from a code sequence
3 group, the plurality of n code sequence registers storing coefficients from one code sequence
4 group of L/n code sequence groups at a time, where L is the number of coefficients in a code
5 sequence, wherein n and L are greater than 1;
6 a plurality of $n+d-1$ sample registers that store sample values from a plurality of d sample
7 sequences that are processed in parallel, wherein d is greater than 1; and
8 a processing unit that processes coefficients in each of the plurality of code sequence
9 groups in the plurality of n code sequence registers in parallel with corresponding sample values
10 in corresponding sample sequence groups from a first plurality of d sample sequences in the
11 plurality of $n+d-1$ sample registers, where each of the code sequence groups is processed to
12 generate intermediate correlation values during a different clock cycle, wherein d may be any
13 value;
14 an accumulation unit that determines a correlation output for each of the sample
15 sequences from the intermediate correlation values generated during the different clock cycles;
16 and
17 a correlation output processor to determine a synchronization point that identifies an
18 amount of delay incurred from transmission of the sample sequences from the correlation output.

1 40. (Previously Presented) The correlator unit of Claim 39, wherein the accumulation
2 unit further comprises an accumulation sub-unit, corresponding to each of the d sample
3 sequences that are processed in parallel, that receives results from the processing unit for a
4 designated sample sequence and that determines a correlation output for the designated sample
5 sequence after each of the code sequence groups are processed.

1 41. (Original) The correlator unit of Claim 39, wherein the processing unit processes the
2 coefficients in each of the plurality code sequence groups in the plurality of n code sequence

3 registers in parallel with corresponding sample values in corresponding sample sequence groups
4 from a second plurality of d sample sequences in the plurality of $n+d-1$ sample registers, where
5 each of the code sequence groups is processed during a different clock cycle.

1 42. (Original) The correlator unit of Claim 40, further comprising correlation output
2 processor that determines a synchronization point for the code sequence from the correlation
3 outputs.

1 43. (Original) The correlator unit of Claim 42, wherein the correlation output processor
2 determines a synchronization point from a correlation output having a highest numerical value.

1 44. (Original) The correlator unit of Claim 39, wherein the processing unit determines a
2 sum of products of the coefficients in each of the code sequence groups with each of the sample
3 values in corresponding sample sequence groups from the first set of d correlation sequences.

1 45. (Original) The correlator unit of Claim 39, wherein the processing is completed after
2 L/n clocks.

1 46. (Previously Presented) A correlator unit, comprising:
2 means for storing coefficients from a code sequence group having n coefficients, the
3 means for storing coefficients from one code sequence group of L/n code sequence groups at a
4 time, where L is a number of coefficients in a code sequence, wherein n and L are greater than 1;
5 means for storing sample values from a plurality of sample sequences that are processed
6 in parallel;
7 means for processing coefficients in each of the plurality of code sequence groups in the
8 means for storing coefficients in parallel with corresponding sample values in corresponding
9 sample sequence groups from a first plurality of contiguous sample sequences in the means for
10 storing sample values, where each of the code sequence groups is processed to generate
11 intermediate correlation values during a different clock cycle;
12 means for determining a correlation output for each of the sample sequences from the
13 intermediate correlation values generated during the different clock cycles; and
14 means for determining a synchronization point that identifies an amount of delay
15 incurred from transmission of the sample sequences from the correlation output.

1 47. (Previously Presented) The method of Claim 1, wherein determining the
2 synchronization point comprises identifying a correlation output having a highest numerical
3 value.

1 48. (Previously Presented) The method of Claim 6, wherein the first group of
2 contiguous sample values are loaded into a set of sample sequence registers during the first clock
3 cycle and the second group of contiguous sample values are loaded into the set of sample
4 sequence registers during the second clock cycle.

1 49. (Previously Presented) The method of Claim 6, wherein determining the
2 synchronization point comprises identifying a correlation output having a highest numerical
3 value.

1 50. (Previously Presented) The method of Claim 14, wherein each of the code sequence
2 groups are loaded into the same set of code sequence registers.